Update on LArSoft Implementation of Space Charge Effect

Michael Mooney

MicroBooNE BNL Meeting

February 11th, 2015

Introduction

- Quick update today on **new** implementation of space charge effect into LArSoft – **simulation only for now**
  - Current implementation for MicroBooNE only
    - Detector-independent implementation coming soon (week?)
  - Temporary hack while database solution is still in progress
  - See MicroBooNE Doc DB **#4071** for previous implementation (since updated) – more details also in backup slides
- Outline:
  - **New** changes to LArSoft (LArEvt, LArSim and uBoone code)
  - How to use new SpaceCharge service for simulating SCE

LArSoft Mods

- Make following changes to LArSoft repositories:
  - **LArEvt**
    - SpaceCharge directory added
    - SpaceCharge/SpaceCharge_service.cc (and SpaceCharge.h)
      - New service to store/access maps for ionization electron displacement due strictly to space charge effect
      - Loads maps from file in constructor, if effect is enabled (see below)
      - Provides accessor methods (**GetPosOffsets**, which returns a vector<double>) to grab displacements given arbitrary {x,y,z} point in TPC – e.g. using **TGraphs** for parametric representation (only option for now)
    - SpaceCharge/spacecharge.fcl
      - FHICL file containing default parameter values for SpaceCharge service
      - Two boolean parameters, **EnableSimulationSCE** and **EnableCorrectionsSCE**, set to **false** by default in **standard_spacecharge** sequence – latter included for **future use only**
      - Two string parameters, **RepresentationType** (e.g. "Parametric") and **InputFilename** ("SCEoffsets.root" by default)
    - SpaceCharge/CMakeLists.txt

UPDATE: Move flag to LArG4Parameters in LArSim

LArSoft Mods (cont.)

- Make following changes to LArSoft repositories:
  - **LArSim**
    - LArG4/LArVoxelReadout.cxx (and LArVoxelReadout.h)
      - Load/utilize **SpaceCharge service** via private member **fSCEHandle**
      - Modify **DriftIonizationElectrons()** to include SCE offsets to individual ionization electron cluster drift, if effect is enabled – checked via member function of SpaceCharge service, **EnableSimulationSCE()**

```
// Get SCE {x,y,z} offsets for particular location in TPC
std::vector<double> posOffsets;
if (fEnableSCE == true)
  posOffsets = fSCEHandle->GetPosOffsets(stepMidPoint.x()/cm,stepMidPoint.y()/cm,stepMidPoint.z()/cm);
else
  posOffsets.resize(3,0.0);

if (tpcg.DriftDirection() == geo::kNegX)
  posOffsets.at(0) *= -1.0;
```

```
// Drift time (nano-sec)
double TDrift;
XDrift += posOffsets.at(0);
TDrift = XDrift * RecipDriftVel[0];
if (tpcg.Nplanes() == 2){// special case for ArgoNeuT (plane 0 is the second wire plane)
  TDrift = ((XDrift - tpcg.PlanePitch(0,1)) * RecipDriftVel[0]
            + tpcg.PlanePitch(0,1) * RecipDriftVel[1]);
}
```

```
// Smear drift times by x position and drift time
G4RandGauss::shootArray( nClus, &XDiff[0], 0., LDiffSig);

// Smear the Y,Z position by the transverse diffusion
G4RandGauss::shootArray( nClus, &YDiff[0], (stepMidPoint.y()/cm)+posOffsets.at(1),TDiffSig);
G4RandGauss::shootArray( nClus, &ZDiff[0], (stepMidPoint.z()/cm)+posOffsets.at(2),TDiffSig);
```

LArSoft Mods (cont.)

- Make following changes to LArSoft repositories:
  - **LArSim (cont.)**
    - LArG4/CMakeLists.txt
  - **uBoone code**
    - uboone/SpaceCharge directory added
    - uboone/SpaceCharge/spacecharge_microboone.fcl
      - Add **microboone_spacecharge** sequence, which is a copy of standard_spacecharge sequence
    - uboone/Utilities/services_microboone_basic.fcl
      - Add **microboone_spacecharge** sequence to basic services list
- Also add SCEoffsets.root file containing TGraphs to $LARSOFT_DATA_DIR
  - Move to $UBOONE_DATA_DIR later and replace with dummy?
- Hack uses **MicroBooNE-specific coordinate transform** in SpaceCharge service's GetPosOffsetsParametric – will update

Using SCE Service

- Extremely simple addition in driver FHICL file:
  - **services.user.LArG4Parameters.EnableSCE: true**
- Can make edits to SpaceCharge service parameters if desired (defaults listed below – no reason to change for now):
  - **services.user.SpaceCharge.RepresentationType: "Parametric"**
  - **services.user.SpaceCharge.InputFilename: "SCEoffsets.root"**

Summary

- Successfully implemented space charge effect into LArSoft
  - Requires modifications to LArEvt, LArSim, uBoone code, and storage of one file in $LARSOFT_DATA_DIR (“**SCEoffsets.root**”)
  - Minor modification may be needed for including other experiments, but this is supposed to be a temporary hack, maybe not necessary
  - Additional G4 runtime with current implementation: **+25%**
  - To be replaced with faster database solution (Randy)

- Code located in **feature/mrmooney_SpaceChargeTest**
  - Already merged into develop in all repositories (same branch name)
  - SCE on: **services.user.LArG4Parameters.EnableSCE: true**

BACKUP SLIDES

Simulation of SC Effect

- Using dedicated code to produce ionization electron displacement maps
  - **Forward transportation**: $\{x, y, z\}_{\mathbf{true}} \rightarrow \{x, y, z\}_{\mathbf{reco}}$
    - Use to **simulate** effect in MC
    - Uncertainties describe accuracy of simulation
  - **Backward transportation**: $\{x, y, z\}_{\mathbf{reco}} \rightarrow \{x, y, z\}_{\mathbf{true}}$
    - Derive from **calibration** and use in data or MC to correct reconstruction bias
    - Uncertainties describe remainder systematic after bias-correction
- Two principal reprexentations to encode displacement maps:
  - **Matrix** – more generic/flexible
  - **Parametric** (5^{th}/7^{th} order polynomials) – fewer parameters
    - Uses matrix representation as input
    - Use in current **LArSoft implementation**

SCE Offsets (Matrix)

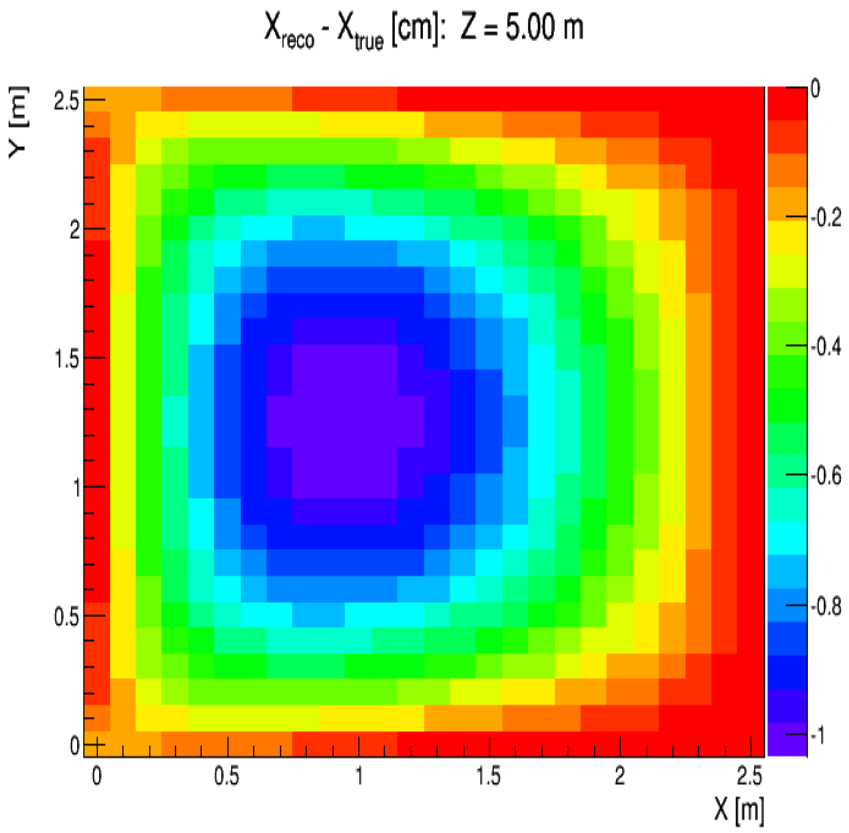
Δx

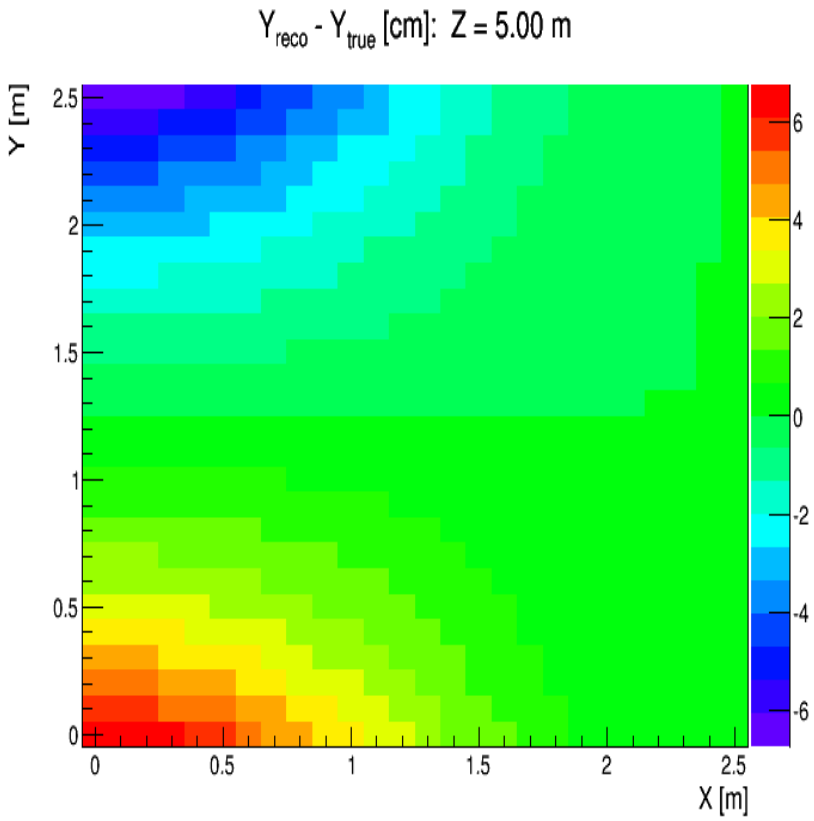
FORWARD

Central
z slice
(Δz not shown)

Δy

FORWARD

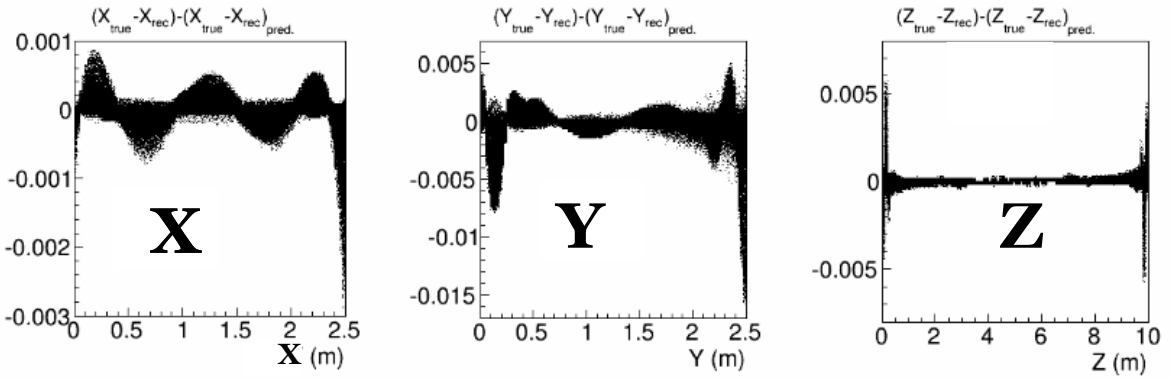




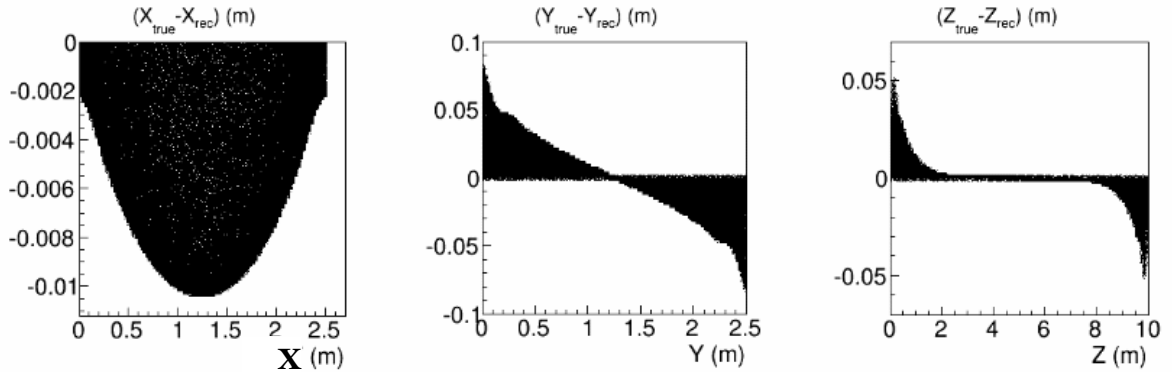
SCE Offsets (Parametric)

LArSoft Implementation

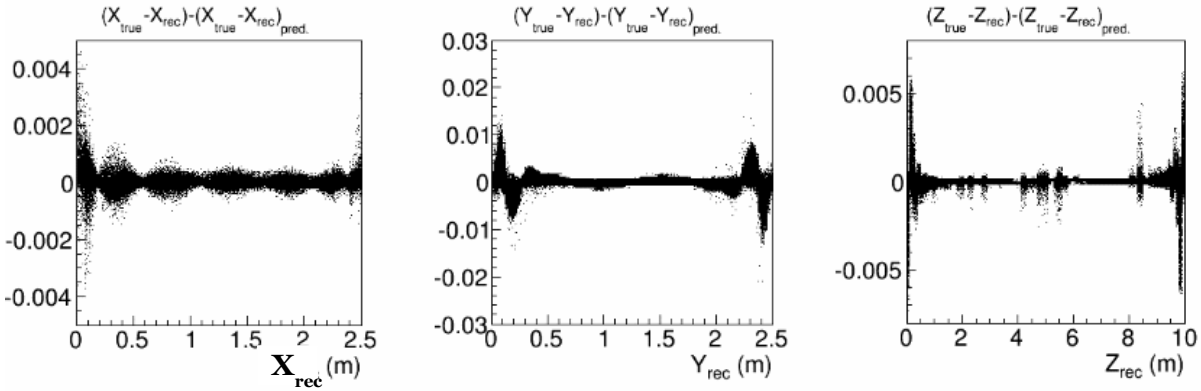
Residuals of Forward Transportation (Uncert. in Simulation of Effect)



Magnitude of Forward Transportation (Reconstruction Bias)

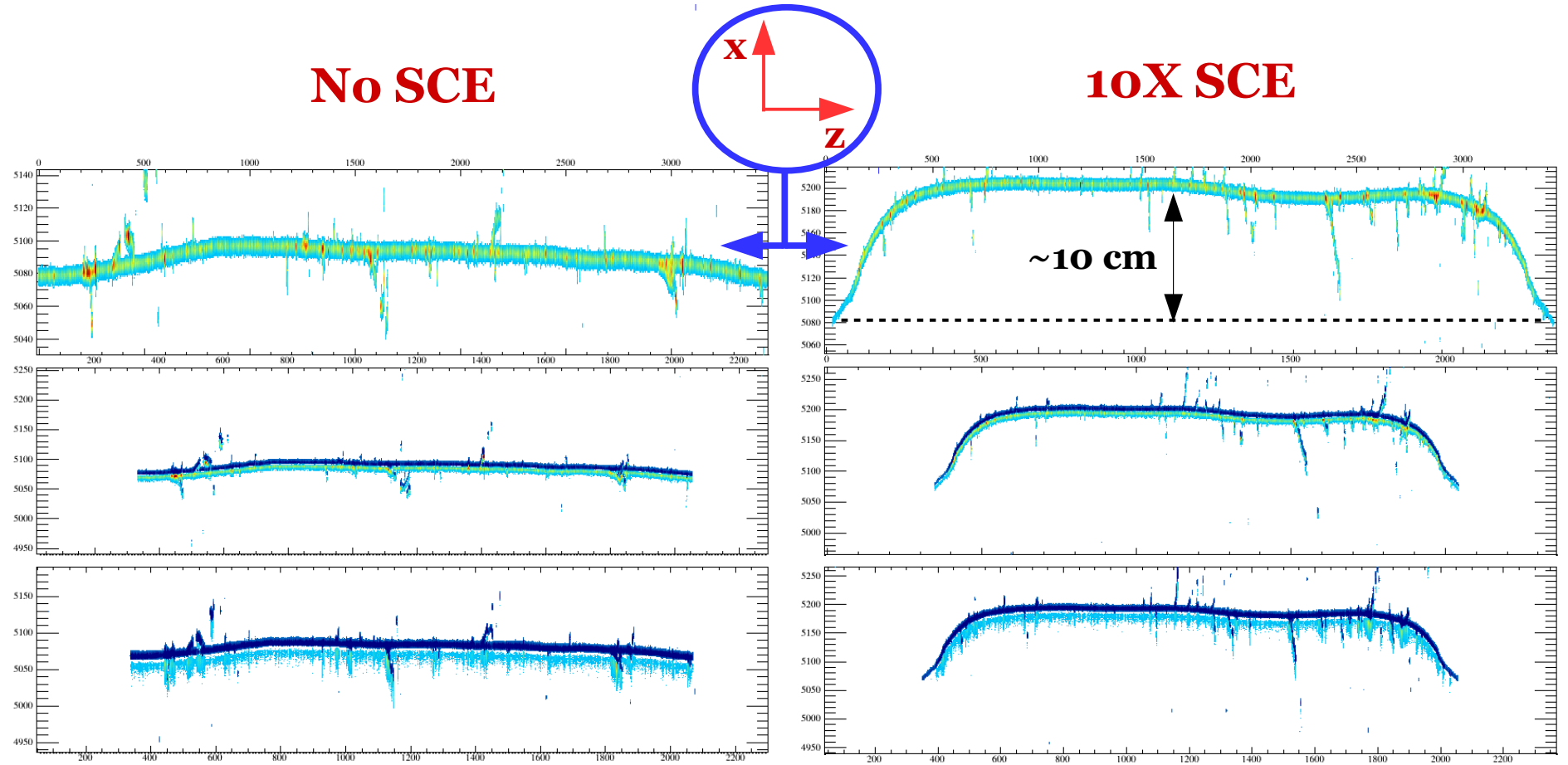


Residuals of Backward Transportation (Post-bias-correction Uncert. for Perfect Calibration)



Validation: SingleGen

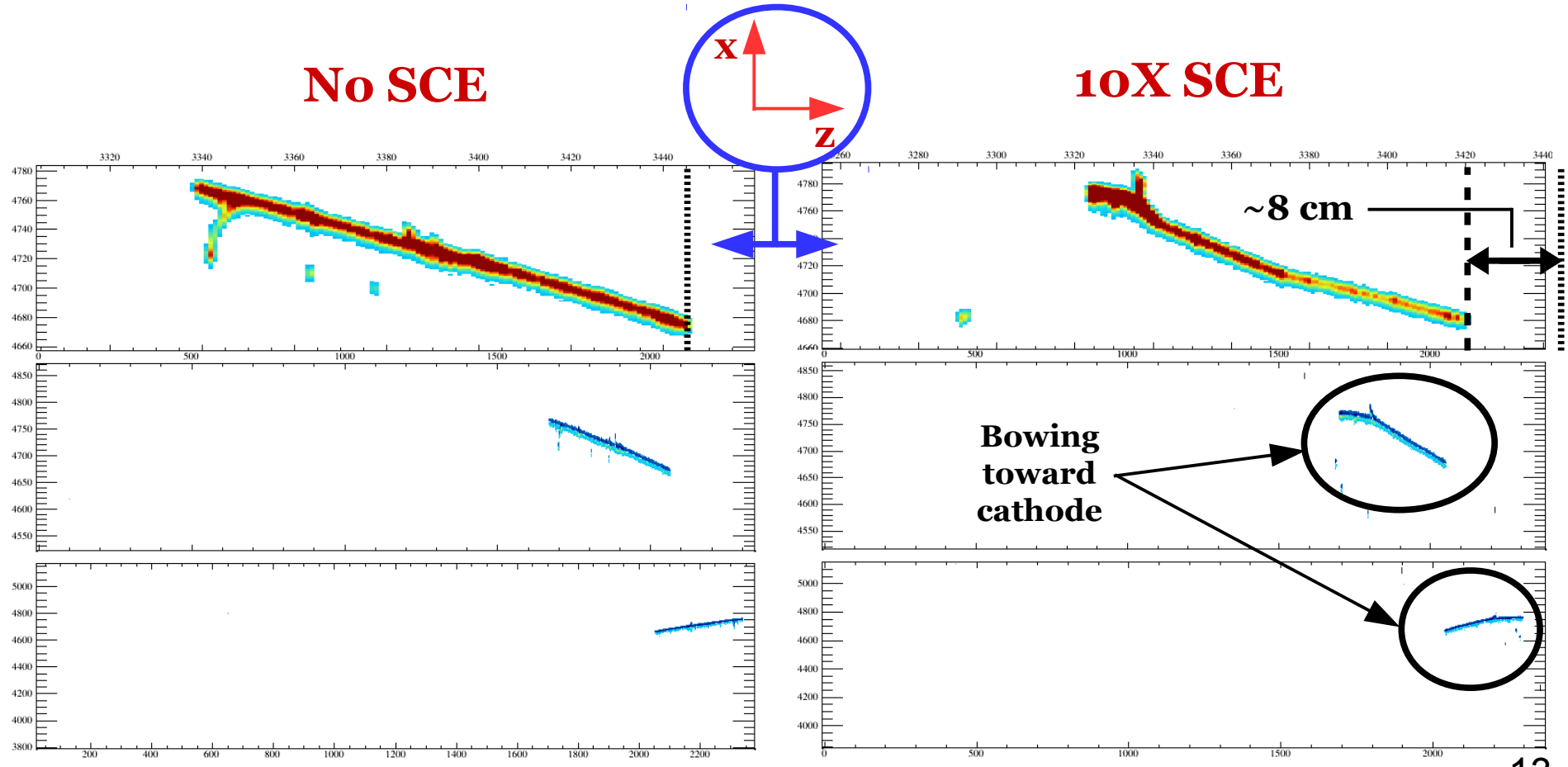
- Produce private MC – 10 GeV muons going through TPC center
- Compare nominal drift to **10X** SCE – seems to be working fine!



Thanks to J. Joshi for technical help

Validation: μBooNE MCC5

- Look at 0.1-2.0 GeV isotropic muon events from MCC5
- Again compare nominal drift to **10X** SCE – no problems!



Thanks to J. Joshi for technical help